> the method comprising the steps of:
>
> receiving an augmented one of the messages (401) from the other component, the other component having augmented the message by adding protocol state information (405) to the message, the protocol state information indicating a state of the other component that is relevant to the protocol;
>
> retaining the state of the other component indicated in the augmented message (413); and
>
> using the retained state to optimize the protocol.

By MPEP 2164.04, Examiner has the "initial burden to establish a reasonable basis to question the enablement provided for the claimed invention." To meet this burden, Examiner states, "The Specification does not contain a clear and concise description such that a skilled artisan can make and use the present invention because 'retained state' is not even mentioned in the Specification".

One problem with the rejection is that "retained state" is mentioned in the Specification as filed, namely in the *Summary of the Invention*:

> The problem of optimizing the two-stage commit protocol is solved as follows: in general, when an action is carried out in a distributed system, a component of the distributed system that is involved in the action is a coordinator for the action and other components involved in the action are cohorts for the action. During the action, the cohorts send messages which are available to the coordinator. In the optimization, each cohort augments messages which are available to the coordinator with information which indicates relevant state of the cohort with regard to the action. The coordinator reads the messages and *retains the most recent relevant state* for each cohort and performs an action according to the relevant state.
>
> When the action is a transaction and the action performed by the coordinator is performing a protocol to ensure that the results of the transaction are consistent in the cohorts, the relevant state indicates whether the transaction will modify data in the cohort and the coordinator optimizes the protocol as determined by the *retained state.* (emphasis added)

As would be expected by its use in the *Summary of the Invention*, the term "retained state" also appears as "retained relevant state" in claim 1 as filed.

A more serious problem is that page 18, line 1 through page 19, line 31, together with FIGs. 4 and 5, provide a complete disclosure of what is claimed in claim 11. As set forth in the discussion of claim 11 at page 3 of Applicants' Appeal Brief,

> In the two-phase commit protocol, which is a species of what is claimed [in claim 11], the protocol state information is that the cohort is read only and the mechanism set forth in the claim for ensuring that the coordinator always knows whether a transaction is read-only with regard to a cohort is described at page 17, line 26-page 18, line 30. The cohorts *augment* the messages 403 belonging to the transaction with protocol state information 405 indicating a state of the cohort that is relevant to the protocol, in this case that the cohort is read only with respect to the transaction. The coordinator keeps track of the state for each cohort as shown at 413, updating a cohort's state as augmented messages 403 come in from the cohort. When it is time to change state, the coordinator can use the retained state for the cohorts to optimize the protocol that is performed when the state changes.
>
> FIG. 5 is a flowchart for using augmented messages 403 indicating (407) whether a cohort is read only with regard to a transaction to optimize the two-phase commit protocol. The flowchart is described at page 19, lines 12-31. The claimed step of "receiving an augmented one of the messages is shown at 515; the step of "retaining the state" is shown at 517; the step of "using the retained state to optimize the protocol" (here, the two-phase commit protocol) is shown at 519, 521, 527, and 529. If retained state 413 indicates that a cohort is read-only with regard to the transaction, the coordinator simply sends the cohort a 2-phase commit "abort" message and doesn't bother with the remainder of the 2-phase commit for that cohort.

The foregoing discussion and FIGs. 4 and 5 provide a complete enablement for the method of claim 11, and since that is the case, the rejection of the claim for lack of enablement is without basis.

The rejection of claim 11 as in the alternative indefinite under 35 U.S.C. 112, second paragraph or not addressed to patentable subject matter under 35 U.S.C. 101

As is clear from Examiner's arguments, this pair of rejections is based on MPEP 2173.05(q) "Use" Claims. The first sentence of MPEP 2173.05(q) states,

> Attempts to claim a process without setting forth any steps involved in the process generally raises an issue of indefiniteness under 35 U.S.C. 112, second paragraph.

MPEP 2173.05 then goes on to state,

> Other decisions suggest that a more appropriate basis for this type of rejection is 35 U.S.C. 101.

Under either theory, the problem is that the process being claimed in claim 11 has three steps and consequently simply does not fall within the ambit of MPEP 2173.05(q). The "method of optimizing a protocol" set forth in claim 11 is concerned with gathering information for use in optimization, not with how the information is actually used. Since that is the case, there is no need for the use to be set forth in any particular detail in claim 11.

The rejection of claim 11 under 35 U.S.C 101 on the basis that the claimed process does not produce a "useful, concrete, and tangible" result

This rejection is based on the recently-issued *Interim Guidelines for Examination of Patent Applications for Patent Subject Matter Eligibility*. In his rejection, Examiner first states that the invention lacks "real world value" because there is no final result.

A number of responses may be made to this rejection:

- The two-phase commit protocol may be used in any situation in which transactions are performed in a distributed system. It constantly has "real world value", and because it has "real world value", so do techniques for optimization of the two-phase commit protocol. The claim is of course broader than the two-phase commit protocol, but the claimed techniques obviously have the same kind of "real world value" for any useful protocol that they can be applied to.
- More importantly, the test set forth in the *Interim Guidelines* is not whether the claimed subject matter has "real world value", but whether it produces a "useful, concrete, and tangible result" As set forth above, the result of the method of claim 11 is clearly "useful"; it is also concrete: as may be seen by comparing FIG. 3 with the flowchart of FIG. 5, an optimized protocol has concrete differences from an unoptimized protocol; finally, the results of an optimized protocol are "tangible". It uses fewer resources than an unoptimized protocol.

Because claim 11 sets forth a process that has "useful", "concrete" and "tangible" results, it is addressed to patentable subject matter under 35 U.S.C. 101 and Examiner's rejection of the claim as not being addressed to patentable subject matter is without foundation.

In the rejection under 35 U.S.C. 101, Examiner also complains that the final limitation "using the retained state to optimize the protocol" "cannot be considered to have real world value because it is unclear what is achieved as a result of optimizing the protocol" and then goes on to complain about the fact that the art uses the term "protocol" in many ways and that "augmenting a message by adding protocol state information" does not comply with the art accepted definition of protocol".

The clarity of a claim's language has nothing whatever to do with whether it is directed to patentable subject matter under 35 U.S.C. 101 and consequently, Examiner's complaints about the language are more properly brought up in the context of a rejection under 35 U.S.C. 112, 2. paragraph, than here, but Applicants will deal with them here. First, as already set forth above, optimizing a protocol has the same results as optimizing anything: it takes fewer resources to do the thing being optimized after optimization than it does before optimization. The claim language is consequently perfectly clear. Second, the "two-phase commit protocol" is well known in the art by that name and is clearly an example of a "protocol [that is] employed . . . in making [a] transaction", which is how the term "protocol" is used in Applicants' claim. Third, the messages are not part of the protocol, but as expressly set forth in claim 11, belong to the transaction with which the protocol is employed. The claimed technique augments the transaction messages with "protocol state information", retains that state information at the coordinator, and uses the retained state information to optimize the protocol. The claim thus clearly distinguishes between the "protocol" and the "messages" of the transaction with which the protocol is being employed. The claim therefore "particularly point[s] out and distinctly claim[s] the subject matter which applicant regards as his invention, which is what 35 U.S.C. 112, 2. par. requires.

The rejections under 35 U.S.C. 103

As a consequence of the appeal, Examiner has finally understood that Lampson does not disclose Applicants' "augmented message belonging to a transaction" and consequently cannot anticipate any of Applicants' claims. To supply the "augmented message belonging to a transaction", Examiner cites Ruberg, which, as set forth in the *Abstract*, discloses

> A distributed settings control protocol. One or more embodiments of the invention provide the ability for an application running on a server across a network to modify various settings related to the terminals such as display resolution, audio output configuration (such as volume control or headphones v. speaker), and energy saver procedures.

The cited location in Ruberg is col. 13, lines 30-35, which reads as follows:

> In accordance with one or more embodiments of the invention, the protocol provides for the terminal (or controlled program) to transmit information to the server (or controlling program). This protocol consists of a list of settings in order by key or number and their values.

Somewhat more enlightenment, including examples, is to be had from lines 36 through col. 14, line 6.

> Additionally, a string may be sent that identifies the controlled program for the purpose of determining the meaning of the indexes which contract/policy the controlling program should follow. The transmission may also include a list of flags that indicate which settings are read-only with respect to this protocol (e.g., another mechanism may have determined that the setting is to be read-only either temporarily or permanently). For example, the following message may be transmitted:
>
> Model Info 0xc6<SIZE:16>
>
> <MODELLEN:8><MODEL:8[MODELLEN]>
>
> <SETLEN:8>:<READONLY:1[SETLEN]>[<PAD:8>]
>
> <SETTINGS:16[SETLEN]>[<PAD:16>]
>
> Such a message may be returned in response to a "Device Control" message (described above) and is transmitted to the various services (e.g., applications and servers) who have sent device control messages so that applications and servers can update their setting information. To allow the entire message to be skipped easily, the above command starts with a byte length of the entire message, including the length field (i.e., SIZE:16).
>
> The first part of the above message is the model name of the terminal. The model name may be specified in terms of a class package name (such as a Java class package name) which describes the vendor, the model name, and ends in the version information for the class package. For example, the following message illustrates a model name of a terminal:
>
> com.sun.HID-P0:alpha3:atr
>
> or com.sun.HID:P1:alpha3:atr
>
> The second part of the above message is the model-opaque sate information. There are a maximum of 255 possible settings. The number of settings may be followed by a bitmap of the read-only flags for each setting. For example, if a "1" is in the leftmost bit of the first byte, then the first parameter may be designated as read only. A pad evens out the byte count to an even 16-bit word (e.g., <PAD:8>).
>
> The last part of the message is a list of 16-bit words that have the values of each indexed setting. The entire command is aligned to a longword.

It will be amply clear from the foregoing that a "message" in Ruberg is simply a message that is sent as part of Ruberg's distributed settings control protocols. Ruberg's messages cannot be equated with the messages of Applicants' claims for two reasons:

- They are not augmented by anything.
- They have no relevance for any other protocol. In Applicants' claims, the augmented messages are not sent as *part of* Applicants' protocol, but are rather used to *optimize* the protocol. There is no suggestion whatever that the messages of Ruberg's distributed settings control protocols can be used to optimize *another* protocol.

Since that is the case, the combination of Ruberg and Lampson does not disclose all of the limitations of Applicants' claims, Examiner has consequently not made the *prima facie* case required for a rejection under 35 U.S.C. 103, and the rejection must fall.

Conclusion

Applicants have traversed all of Examiner's grounds of rejection and have consequently been fully responsive to Examiner’s non-final Office action of 9/6/2006 as required by 37 C.F.R. 1.111(b) and respectfully request that Examiner continue with his examination and allow the claims as amended, as provided by 37 C.F.R. 1.111(a). No fees are believed to be required for this response. Should any be, please charge them to Deposit Account Number 501315.

Respectfully submitted,
/Gordon E. Nelson/

Attorney of record,
Gordon E. Nelson
57 Central St., P.O. Box 782
Rowley, MA, 01969,
Registration number 30,093
Voice: (978) 948-7632
Fax: (866) 723-0359
12/6/2006
Date